

**IN THE CLAIMS:**

---

Claim 1 (Currently Amended): A server comprising:

at least one port driver, each port driver corresponding to a port to which a port device can be connected; and,

an automatic plug-and-play component designed to detect connection and disconnection of a port device to a port having a corresponding port driver selected from the at least one port driver, the component without user intervention automatically installing an appropriate device driver for the port device upon connection thereof to the port, such that the port device is accessible by clients communicatively coupled to the server, and automatically uninstalling the appropriate device driver upon disconnection of the port device from the port, such that the port device is inaccessible by the clients.

Claim 2 (Original): The server of claim 1, wherein the port device is one of: a printer, a scanner, a fax machine, a multi-function device (MFD), and a digital camera.

Claim 3 (Original): The server of claim 1, wherein the port is a parallel port.

Claim 4 (Original): The server of claim 1, wherein the port is a serial port.

Claim 5 (Original): The server of claim 4, wherein the serial port has a Universal Serial Bus (USB) form factor.

Claim 6 (Original): The server of claim 4, wherein the serial port has an IEEE1394 form factor.

Claim 7 (Currently Amended): The server of claim 1, wherein upon connection of the port device to the port, the automatic plug-and-play component retrieves a plug-and-play identifier from the port device, and selects the appropriate device driver based on the plug-and-play identifier.

Claim 8 (Currently Amended): The server of claim 1, wherein installing the appropriate device driver for the port device includes downloading the appropriate driver from the Internet.

Claim 9 (Original): The server of claim 1, wherein the server is a server appliance lacking at least a dedicated keyboard and a dedicated monitor.

AL  
Claim 10 (Currently Amended): A machine-readable medium having stored thereon a computer program for execution by a processor of a server communicatively coupled to one or more clients, the computer program comprising:

at least one port class driver, each port class driver designed to pass signals to and receive signals from a corresponding port driver corresponding to a port to which a port device can be connected; and,

a monitor designed to monitor connection and disconnection of a port device to a port selected from the at least one port driver, the ~~logic~~ monitor causing without user intervention automatic installation of an appropriate device driver for the port device upon connection thereof to the port, such that the device is accessible by the clients, and automatic uninstallation of the appropriate device driver upon disconnection of the port device from the port, such that the port device is inaccessible by the clients.

Claim 11 (Original): The medium of claim 10, wherein the port device is one of: a printer, a scanner, a fax machine, a multi-function device (MFD), and a digital camera.

Claim 12 (Original): The medium of claim 10, wherein the port is a parallel port.

Claim 13 (Original): The medium of claim 10, wherein the port is a serial port.

Claim 14 (Original): The medium of claim 10, wherein the monitor is implemented as a service.

Claim 15 (Original): The medium of claim 10, wherein the monitor resides in user mode, and each of the at least one port class driver resides in a kernel mode.

Claim 16 (Original): The medium of claim 10, wherein the monitor incorporates monitoring logic implemented as a state-transition system.

Ab Claim 17 (Currently Amended): The medium of claim 10, wherein upon connection of the port device to the port, the monitor retrieves a plug-and-play identifier from the port device, and selects the appropriate device driver based on the plug-and-play identifier.

Claim 18 (Original): The medium of claim 10, wherein the program further comprises a support component designed to pass signals from the monitor to the at least one port class driver and vice-versa.

Claim 19 (Original): The medium of claim 18, wherein the support program resides in a user mode.

Claim 20 (Original): The medium of claim 18, wherein the support component comprises a first component and at least one second component, the first component designed to determine a number of ports of the server and instantiate a number of the at least one second component equal to the number of ports, each second component designed to pass signals from the monitor to one of the at least one port class driver and vice-versa.

Claim 21 (Currently Amended): The medium of ~~claim 18~~ claim 20, wherein each of the first component and the at least one second component is implemented as an object.

Claim 22 (Original): The medium of claim 18, wherein the support component is able to pass signals from an external monitor not residing at the server to the at least one port class driver and vice-versa.

Claim 23 (Original): The medium of claim 22, wherein the external monitor is implemented as a web component.

Q6  
Claim 24 (Currently Amended): A machine-readable medium having stored thereon as a computer program for execution by a processor and a monitoring logic for automatic device plug-and-play without user intervention that is implemented as a state transition system comprising:

- a first state in which a device driver is not installed and a plug-and-play identifier has been detected from a device connected to a port;

- a second state in which the device driver is installed and the plug-and-play identifier has been detected from the device connected to the port;

- a third state in which the device driver is not installed and the plug-and-play identifier has not been detected from the device connected to the port; and,

- a fourth state in which the device driver is installed and the plug-and-play identifier has not been detected from the device connected to the port, wherein the states are transitioned among one another based on a set of transitions comprising:

  - a first state-to-second state transition based on a ~~manual~~-device driver installation event;

  - a first state-to-third state transition based on an event of no detection of the plug-and-play identifier;

  - a second state-to-first state transition based on a ~~manual~~-device driver uninstallation event;

  - a second state-to-second state transition based on an event of detection of a new plug-and-play identifier;

  - a second state-to-fourth state transition based on the event of no detection of the plug-and-play identifier;

a third state-to-second state transition based on an event of detection of the plug-and-play identifier;

a third state-to-fourth state transition based on the ~~manual~~ device driver installation event;

a fourth state-to-third state transition based on the ~~manual~~ device driver uninstallation event; and,

a fourth state-to-second state transition based on the event of detection of the plug-and-play identifier.

---